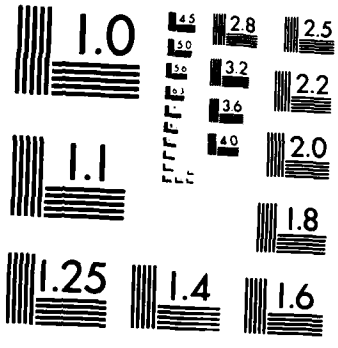
AD-A147 237 A DUAL QUADRATIC PROGRAMMING ALGORITHM(U) WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER K RITTER AUG 84 MRC-TSR-2733 DAAG29-80-C-0041 1/1

UNCLASSIFIED F/G 12/1 NL



(3)

MRC Technical Summary Report #2733

A DUAL QUADRATIC PROGRAMMING ALGORITHM

Klaus Ritter

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53705

(Received April 27, 1984)

August 1984

Approved for public release
Distribution unlimited

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

UNIVERSITY OF WISCONSIN-MADISON
MATHEMATICS RESEARCH CENTER

A DUAL QUADRATIC PROGRAMMING ALGORITHM

Klaus Ritter

Technical Summary Report #2733
August 1984

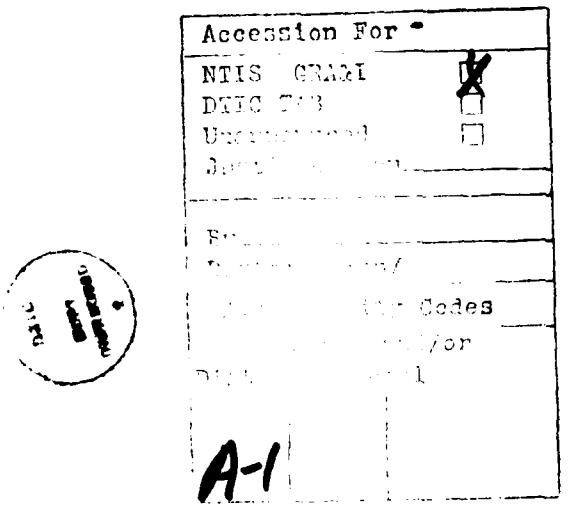
ABSTRACT

By using conjugate directions a method for solving convex quadratic programming problems is developed. The algorithm generates a sequence of dual feasible solutions and terminates after a finite number of steps.

AMS (MOS) Subject Classifications: 90C20, 90C25

Key Words: Quadratic programming, Duality, Optimization, Conjugate directions

Work Unit Number 5 (Optimization and Large Scale Systems)



Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.

SIGNIFICANCE AND EXPLANATION

The quadratic programming problem is the following: Given $n \times 1$ vectors $c, a_1, \ldots, a_m$, numbers $b_1, \ldots, b_m$ and an $n \times n$ matrix C, find an $n \times 1$ vector x which minimizes the quadratic function

$$c'x + \frac{1}{2} x'Cx ,$$

subject to the inequality constraints

$$a_i'x \leq b_i; \quad i = 1, \ldots, m .$$

If C is the $n \times n$ zero matrix, then the quadratic programming problem reduces to the linear programming problem.

In recent years quadratic programming has become an important tool in optimization. It has wide applications in areas such as statistics, structural engineering, economics and portfolio analysis.

In applications the following situation occurs frequently: A quadratic programming problem has been formulated and an optimal solution x has been obtained. Then it is discovered that additional constraints are required which render the previous optimal solution x unfeasible. The contribution of this work is an algorithm which solves the new problem in an efficient way by taking advantage of the fact that an optimal solution x to the "wrong" problem is known.

A DUAL QUADRATIC PROGRAMMING ALGORITHM

Klaus Ritter

1) Introduction

In mathematical programming the relationship between a primal minimization problem and the associated dual maximization problem plays an important role. In particular, if an optimal solution to either problem is known, an optimal solution to the other problem can easily be obtained. Most algorithms for solving the primal problem generate a sequence of primal feasible solutions with decreasing objective function values. On the other hand a dual method determines a sequence of dual feasible solutions with increasing values of the primal objective function. Only the optimal solution is primal feasible.

For quadratic programming problems such a dual method is developed in this paper. In the next section we introduce some definitions and preliminary results. Section 3 contains an outline of the algorithm and its properties. In the final section a detailed description of the algorithm is given.

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.

2) Formulation of the Problem and Basic Results

We consider the problem

$$\min\{c'x + \frac{1}{2} x'Cx \mid Ax \leq b\} , \tag{2.1}$$

where c and x are n-dimensional vectors, b is an m-dimensional vector and A is an (m,n)-matrix. Thus n denotes the number of variables and m denotes the number of inequality constraints.

Throughout the paper we assume that the (n,n)-matrix C is symmetric and positive definite. This implies that the objective function $c'x + \frac{1}{2} x'Cx$ is strictly convex. Therefore, (2.1) has a unique optimal solution if the set

$$R = \{x \mid Ax \leq b\}$$

of feasible solutions is non-empty. With

$$Q(x) = c'x + \frac{1}{2} x'Cx, \quad A' = (a_1,\ldots,a_m), \quad b' = (b_1,\ldots,b_m)$$

we can write (2.1) in the equivalent form

$$\min\{Q(x) \mid a_i'x \leq b_i, \quad i = 1,\ldots,m\}.$$

To avoid some technical difficulties we assume that, for each $x \in R$, the gradients of the constraints active at x are linearly independent.

Problem (2.1) is sometimes called the primal problem and associated with the following maximization problem which is said to be the dual problem

$$\max\{-b'u - \frac{1}{2} x'Cx \mid Cx + A'u = -c, \; u \geq 0\} . \tag{2.2}$$

Here u is an m-dimensional vector. Thus the dual problem has $n + m$ variables. The main results connecting the two problems are the duality theorems which assert that if $\hat{x}$ is an optimal solution to the primal problem, then there is $\hat{u}$ such that the pair $(\hat{x},\hat{u})$ is an optimal solution to the dual problem and, vice versa if $(\tilde{x},\tilde{u})$ is an optimal solution to the dual problem then $\tilde{x}$ is an optimal solution to the primal problem (see e.g. [2]).

If $Q(x)$ is convex it is well-known (see e.g. [2]) that the following optimality conditions are necessary and sufficient for an optimal solution of (2.1).

Optimality conditions

$\hat{x}$ is an optimal solution to (2.1) if and only if there is $\hat{u} = (\hat{u}_1,\ldots,\hat{u}_m)'$ such that

i) $C\hat{x} + A'\hat{u} = -c, \quad \hat{u} \geq 0$,

ii) $A\hat{x} \leq b$,

iii) $\hat{u}_i(a_i'\hat{x} - b_i) = 0, \quad i = 1,\ldots,m$.

An $\hat{x}$ is said to be a quasi-stationary point if there is some $\hat{u}$ such that $(\hat{x},\hat{u})$ satisfies the optimality conditions with the possible exception of the non-negativity condition $\hat{u} \geq 0$.

Let $\hat{x}$ be a given stationary point and assume that

$$a_i'\hat{x} = b_i, \quad i = 1,\ldots,p, \qquad a_i'\hat{x} < b_i, \quad i = p+1,\ldots,m .$$

Then

$$-c - C\hat{x} = \sum_{i=1}^{p} \hat{u}_i a_i$$

and for every

$$x \in \{x \mid a_i'x = b_i, \quad i = 1,\ldots,p\}$$

we have

$$\begin{aligned} Q(x) &= Q(\hat{x}) - (c + C\hat{x})'(\hat{x} - x) + \frac{1}{2}(\hat{x} - x)'C(\hat{x} - x) \\ &= Q(\hat{x}) + \frac{1}{2}(\hat{x} - x)'C(\hat{x} - x) . \end{aligned}$$

This shows that $\hat{x}$ is the unique optimal solution to the problem

$$\min\{Q(x) \mid a_i'x = b_i, \quad i = 1,\ldots,p\} .$$

Therefore, problem (2.1) has only finitely many quasi-stationary points and every algorithm which generates a sequence of quasi-stationary points $\hat{x}_1,\hat{x}_2,\ldots$ with the property

$$Q(\hat{x}_{j+1}) < Q(\hat{x}_j)$$

will terminate with the optimal solution to (2.1) after a finite number of iterations.

It does not seem to be practical to construct algorithms which obtain a new quasi-stationary point at every iteration. However, a typical primal method for solving (2.1) will generate a sequence of feasible solutions x_j with the following properties.

i) Every subsequence of $n + 1$ consecutive points $x_\nu, x_{\nu+1}, \ldots, x_{\nu+n}$ contains at least one quasi-stationary point.

ii) If x_{j_1} and x_{j_2} are two quasi-stationary points with $j_1 < j_2$, then

$$Q(x_{j_2}) < Q(x_{j_1}) .$$

An example of such an algorithm is given in [1].

An $\hat{x}$ is said to be a pseudo-stationary point if there is some $\hat{u}$ such that $(\hat{x},\hat{u})$ satisfies the optimality conditions with the possible exception of some of the primal feasibility conditions $a_i'x \leq b_i$. In practice such a point is obtained if, after solving (2.1), it is discovered that more constraints have to be added which render the present optimal solution $\hat{x}$ infeasible.

Let $\hat{x}$ be a pseudo-stationary point and assume that

$$a_i'\hat{x} = b_i, \quad i = 1,\ldots,p, \qquad a_i'\hat{x} \neq b_i, \quad i = p+1,\ldots,m .$$

Then

$$-c - C\hat{x} = \sum_{i=1}^{p} \hat{u}_i a_i, \quad \hat{u}_i > a, \quad i = 1,\ldots,p .$$

Thus it follows from the optimality conditions that $\hat{x}$ is the unique optimal solution to the problem

$$\min\{Q(x) \mid a_i'x \leq b_i, \quad i = 1,\ldots,p\} . \tag{2.3}$$

This shows that (2.1) has only finitely many pseudo-stationary points.

Clearly, a pseudo-stationary point is the optimal solution to (2.1) if and only if it is feasible. Let $\hat{x}$ be the optimal solution to (2.1) and let $\hat{x}_j$ be any pseudo-stationary point. Then it follows from (2.3) that either $\hat{x} = \hat{x}_j$ or $Q(\hat{x}) > Q(\hat{x}_j)$. Thus any algorithm which generates a sequence of pseudo-stationary points $\hat{x}_1, \hat{x}_2, \ldots$ with

$$Q(\hat{x}_{j+1}) > Q(\hat{x}_j)$$

will terminate after a finite number of iterations. As in the case of primal algorithms it is not practical to insist that the algorithm obtains a new pseudo-stationary point at every iteration. Instead, we will develop an algorithm which generates a sequence $\{x_j, u_j\}$ of dual feasible solutions with the following properties.

i) Every subsequence $\{x_\nu, x_{\nu+1}, \ldots, x_{\nu+n}\}$ contains at least one pseudo-stationary point.

ii) If x_{j_1} and x_{j_2} are two pseudo-stationary points with $j_1 < j_2$, then

$$Q(x_{j_2}) > Q(x_{j_1}) .$$

3) A General Outline of the Algorithm

Let x_j be a pseudo-stationary point and assume that

$$a_i^t x_j = b_i, \quad i = 1,\ldots,p .$$

Then there are numbers $u_{1j},\ldots,u_{pj}$ such that

$$Cx_j + \sum_{i=1}^{p} u_{ij}a_i = -c, \quad u_{ij} \geq 0$$

$$a_i^t x_j = b_i, \quad i = 1,\ldots,p . \tag{3.1}$$

If x_j satisfies the inequalities $a_i^t x \leq b_i$ for $i = p+1,\ldots,m$, then it follows from the optimality conditions that x_j is an optimal solution. Let k be any integer such that

$$a_k^t x_j > b_k . \tag{3.2}$$

First we will determine a pseudo-stationary point for which the kth constraint is satisfied as an equality. There are two cases to be considered depending on whether a_k is linearly dependent on $a_1,\ldots,a_p$.

Case 1: $a_k \notin \text{span}\{a_1,\ldots,a_p\}$.

Then the equations $a_i x = b_i$, $i = 1,\ldots,p$, $i = k$ are consistent and the problem

$$\min\{Q(x) \mid a_i x = b_i, \quad i = 1,\ldots,p, \quad i = k\} \tag{3.3}$$

has a unique optimal solution which we denote by y_j. Furthermore, there are numbers v_{ij} such that

$$Cy_j + \sum_{i=1}^{p} v_{ij}a_i + v_{kj}a_k = -c ,$$

$$a_i^t y_j = b_i, \quad i = 1,\ldots,p , \tag{3.4}$$

$$a_k^t y_j = b_k .$$

Setting $s_j = x_j - y_j$, $u_{kj} = 0$,

$$\delta_{kj} = u_{kj} - v_{kj} \quad \text{and} \quad \delta_{ij} = u_{ij} - v_{ij}, \quad i = 1,\ldots,p$$

we obtain from (3.1) and (3.4) the equalities

$$Cs_j + \sum_{i=1}^{p} \delta_{ij} a_i + \delta_{kj} a_k = 0 \tag{3.5}$$

$$a_i' s_j = 0, \quad i = 1,\ldots,p \tag{3.6}$$

$$a_k' s_j = a_k' x_j - b_k \, . \tag{3.7}$$

Furthermore, let

$$\sigma_j = \min\{1, \frac{u_{ij}}{\delta_{ij}} \mid \text{for all } i \text{ with } \delta_{ij} > 0\} \tag{3.8}$$

and set

$$x_{j+1} = x_j - \sigma_j s_j, \; u_{k,j+1} = u_{kj} - \sigma_j \delta_{kj}, \; u_{i,j+1} = u_{ij} - \sigma_j \delta_{ij}, \quad i = 1,\ldots,p \, .$$

Then

$$Q(x_{j+1}) - Q(x_j) = -\sigma_j (c + Cx_j)' s_j + \frac{\sigma_j^2}{2} s_j' C s_j$$

$$= \sigma_j \sum_{i=1}^{p} u_{ij} a_i' s_j + \frac{\sigma_j^2}{2} s_j' C s_j$$

$$= \frac{\sigma_j^2}{2} s_j' C s_j \, .$$

Multiplying (3.5) by s_j we obtain

$$s_j' C s_j = -\delta_{kj} a_k' s_j \, .$$

Since $s_j' C s_j > 0$ and $a_k' s_j > 0$ we have $-\delta_{kj} > 0$ and

$$u_{k,j+1} = u_{kj} - \sigma_j \delta_{kj} \geq u_{kj} \geq 0 \, .$$

Thus, if $\sigma_j = 1$, it follows that $x_{j+1} = x_j - s_j$ is a pseudo-stationary point with

$$Q(x_{j+1}) < Q(x_j) \quad \text{and} \quad a_k' x_{j+1} = b_k \, .$$

If $\sigma_j < 1$, let $\sigma_j = u_{\ell j}/\delta_{\ell j}$, say. Then $u_{\ell,j+1} = 0$ and $a_k' x_{j+1} > b_k$. Replacing x_j with x_{j+1} and deleting $a_\ell' x = b_\ell$ from the set of active constraints we can repeat the above steps to obtain

$$x_{j+2} = x_{j+1} - \sigma_{j+1} s_{j+1}, \; u_{k,j+2} \geq 0, \; u_{i,j+2} \geq 0, \quad i = 1,\ldots,p, \quad i \neq \ell \, .$$

If $\sigma_{j+1} = 1$, then x_{j+2} is a pseudo-stationary point with

$$Q(x_{j+2}) > Q(x_j) \quad \text{and} \quad a_k' x_{j+2} = b_k \ .$$

If $\sigma_{j+1} < 1$, then $a_k' x_{j+2} > b_k$ and $u_{i,j+2} = 0$ for at least one $i = 1,\ldots,p,\ i \neq \ell$.

Repeating these steps we obtain an $x_{j+\rho}$ with $1 \leq \rho \leq n$ such that $x_{j+\rho}$ is a pseudo-stationary point with

$$Q(x_{j+\rho}) > Q(x_j) \quad \text{and} \quad a_k' x_{j+\rho} = b_k \ . \tag{3.9}$$

Case 2: $a_k = \lambda_1 a_1 + \cdots + \lambda_p a_p$

Because $a_k' x_j > b_k$, (3.3) has no feasible solutions. If $\lambda_i \leq 0$ for all $i = 1,\ldots,p$, then the given problem (2.1) has no feasible solutions as will be shown in the proof of the theorem in the next section. If at least one λ_i is positive, determine the smallest index ν such that

$$\frac{u_{\nu j}}{\lambda_\nu} = \min\left\{\frac{u_{ij}}{\lambda_i} \;\middle|\; \text{for all } i \text{ with } \lambda_i > 0\right\} \tag{3.10}$$

and set

$$\hat{u}_{kj} = \frac{u_{\nu j}}{\lambda_\nu}, \quad \hat{u}_{ij} = u_{ij} - \frac{u_{\nu j}}{\lambda_\nu}\lambda_i, \quad i = 1,\ldots,p \ .$$

Replacing (3.1) with

$$Cx_j + \sum_{\substack{i=1 \\ i\neq\nu}}^{p} \hat{u}_{ij} a_i + \hat{u}_{kj} a_k = -c$$

$$a_i' x_j = b_i, \quad i = 1,\ldots,p, \quad i \neq \ell$$

and (3.3) with

$$\min\{Q(x) \mid a_i' x = b_i, \ i = 1,\ldots,p, \ i = k, \ i \neq \nu\}$$

we can proceed as in Case 1 to obtain a pseudo-stationary point $x_{j+\rho}$ with the properties (3.9).

The above description of the algorithm indicates that all data required to perform one iteration can be obtained by solving a system of linear equations of the form (3.5)-(3.7). Depending on the number of active constraints the dimension of this system varies from n to $2n$. In the following we will show that these data can also be derived from an appropriate (n,n)-matrix associated with x_j.

Let x_j be again a pseudo-stationary point with

$$a_i^t x_j = b_i, \quad i = 1,\ldots,p$$

and define

$$T_j = \{x \mid a_i^t x = 0, \quad i = 1,\ldots,p\} .$$

Since the gradients of active constraints are assumed to be linearly independent the subspace T_j has dimension $n - p$. Let $c_{p+1,j},\ldots,c_{nj}$ be a basis of conjugate directions for T_j. By this we mean the following.

i) $c_{p+1,j},\ldots,c_{nj}$ are linearly independent,

ii) $c_{ij} \in T_j, \quad i = p + 1,\ldots,n$,

iii) $c_{ij}^t C c_{kj} = 0, \quad i,k = p + 1,\ldots,n, \quad i \neq k,$

iv) $c_{ij}^t C c_{ij} = 1, \quad i = p + 1,\ldots,n$.

Because C is positive definite it is not difficult to verify that such a basis for T_j exists.

Now define the (n,n)-matrix D_j^t as follows

$$D_j^t = (a_1,\ldots,a_p, Cc_{p+1,j},\ldots,Cc_{nj}) .$$

Then D_j^t is non-singular and denoting the columns of D_j^{-1} by $c_{1j},\ldots,c_{nj}$ we have

$$D_j^{-1} = (c_{1j},\ldots,c_{pj}, c_{p+1,j},\ldots,c_{nj})$$

where $c_{p+1,j},\ldots,c_{nj}$ are the vectors that form a basis of conjugate directions for T_j.

Since $a_i^t c_{ij} = 1$ and $a_i^t c_{kj} = 0$ for $i,k = 1,\ldots,p$, $i \neq k$, it follows immediately from (3.1) that

$$u_{ij} = -c_{ij}^t(c + Cx_j), \quad i = 1,\ldots,p .$$

Furthermore, $a_k \in \text{span}\{a_1,\ldots,a_p\}$ if and only if

$$a_k^t c_{ij} = 0 \quad \text{for all} \quad i = p + 1,\ldots,n .$$

First assume that $a_1,\ldots,a_p,a_k$ are linearly independent and set

$$\hat{T}_j = \{x \mid a_i^t x = 0, \quad i = 1,\ldots,p, \quad i = k\} .$$

In Step 2.1 of the algorithm given in the next section it is shown how a basis of conjugate directions $c_{p+2,j+1},\ldots,c_{n,j+1}$ for $\hat{T}_j$ can be derived from $c_{p+1,j},\ldots,c_{nj}$. Setting

$$D_{j+1}' = (a_1,\ldots,a_p,a_k,\ Cc_{p+2,j+1},\ldots,Cc_{n,j+1})$$

we obtain

$$D_{j+1}^{-1} = (c_{1,j+1},\ldots,c_{p+1,j+1},\ c_{p+1,j+1},\ldots,c_{n,j+1})$$

where $c_{p+2,j+1},\ldots,c_{n,j+1}$ form a basis of conjugate directions for $\hat{T}_j$.

With

$$s_j = (a_k'x_j - b_k)c_{p+1,j+1} \tag{3.11}$$

it follows from the definition of the inverse matrix that

$$a_i's_j = 0,\quad i = 1,\ldots,p,\quad \text{and}\quad a_k's_j = a_k'x_j - b_k\ .$$

Furthermore, $c_{i,j+1}'Cs_j = 0$, $i = p+2,\ldots,n$ shows that

$$Cs_j \in \text{span}\{a_1,\ldots,a_p,a_k\}\ .$$

Therefore, the solution of the equations (3.5)-(3.7) is given by (3.11) and

$$\delta_{ij} = -c_{i,j+1}'Cs_j,\quad i = 1,\ldots,p,\ i = k\ .$$

Let σ_j be as defined by (3.8) and set $x_{j+1} = x_j - \sigma_j s_j$. If $\sigma_j = 1$, x_{j+1} is a pseudo-stationary point. If $\sigma_j < 1$, let $\sigma_j = u_{\ell j}/\delta_{\ell j}$ and define

$$T_{j+1} = \{x \mid a_i'x = 0,\quad i = 1,\ldots,p,\ i = k,\ i \neq \ell\}\ .$$

Then the vectors

$$c_{\ell,j+2} = \frac{c_{\ell,j+1}}{\sqrt{c_{\ell,j+1}'Cc_{\ell,j+1}}}\ ,$$

$$c_{i,j+2} = c_{i,j+1},\quad i = p+2,\ldots,n$$

form a basis of conjugate directions for T_{j+1}. Assuming for simplicity that $\ell = p-1$ we have

$$D_{j+2}' = (a_1,\ldots,a_{p-1},\ Cc_{\ell,j+2},a_k,Cc_{p+2,j+2},\ldots,Cc_{n,j+2})$$

and

$$D_{j+2}^{-1} = (c_{1,j+2},\ldots,c_{n,j+2})\ .$$

If $a_k \in \text{span}\{a_1,\ldots,a_p\}$, then $\hat{T}_j = T_j$. With ν as defined by (3.10) we have

$$D_{j+1}' = (a_1,\ldots,a_{\nu-1},a_k,a_{\nu+1},\ldots,a_p,Cc_{p+1,j+1},\ldots,Cc_{n,j+1})$$

and

$$D_{j+1}^{-1} \quad (c_{1,j+1},\ldots,c_{n,j+1}) \ ,$$

where

$$c_{i,j+1} = c_{ij} \quad \text{for} \quad i = p + 1,\ldots,n \ .$$

The matrix D_{j+2}^{j} is then determined as in the previous case.

All information needed in an iteration of the algorithm can be derived from the matrix D_j^{-1}. The matrix D_j^{j} is only used to illustrate the definition of D_j^{-1}. In order to differentiate the columns of D_j^{-1} which form a basis for T_j from those which correspond to gradients of active constraints we associate with D_j^{-1} an index set

$$J(x_j) = \{\alpha_{1j},\ldots,\alpha_{nj}\} \ .$$

For each i, $\alpha_{ij} \in \{0,1,\ldots,m\}$ and $\alpha_{ij} = 0$ if and only if $c_{ij} \in T_j$. Furthermore, $\alpha_{ij} = k > 0$ if and only if a_k is in the ith column of D_j^{j}.

4) Detailed Description of the Algorithm

As initial data, the algorithm requires a pseudo-stationary point (x_0,u_0) and the associated matrix $D_0^{-1} = (c_{10},\ldots,c_{n0})$ and index set $J(x_0) = \{\alpha_{10},\ldots,\alpha_{n0}\}$ as defined in the previous section. There are two important cases in which these data are easily available. First, if C is a diagonal matrix, then x_0 can be the unconstrained minimizer of $Q(x)$. All columns of D_0^{-1} are conjugate directions. Second, x_0, u_0, D_0^{-1}, and $J(x_0)$ could be obtained by applying the algorithm given in [1] to the original problem which then was augmented by additional constraints not satisfied by x_0.

A general iteration of the algorithm is as follows.

Step 1

Determine the smallest k such that

$$a_k'x_j - b_k = \max\{a_i'x_j - b_i \mid i = 1,\ldots,m\} .$$

If $a_k'x_j - b_k \leq 0$, stop with the optimal solution x_j; otherwise go to Step 2.1.

Step 2.1

If $\alpha_{ij} > 0$ for all $i = 1,\ldots,n$, go to Step 2.2; otherwise compute

$$z_i = a_k'c_{ij} \quad \text{for all } i \text{ with } \alpha_{ij} = 0 .$$

Determine the smallest ν such that

$$|z_\nu| = \max\{|z_i| \mid \text{for all } i \text{ with } \alpha_{ij} = 0\} .$$

If $z_\nu = 0$, set

$$c_{i,j+1} = c_{ij} \quad \text{for all } i \text{ with } \alpha_{ij} = 0$$

and go to Step 2.2. If $z_\nu \neq 0$, set $z' = (z_1,\ldots,z_n)$ with $z_i = 0$ for all i with $\alpha_{ij} > 0$, $\delta = 1$ if $z_\nu \leq 0$ and $\delta = -1$ if $z_\nu > 0$. Compute

$$w = \frac{\delta\|z\|e_\nu - z}{\|\delta\|z\|e_\nu - z\|} = (w_1,\ldots,w_n)' ,$$

where e_ν is the ν-th unit vector. Set

$$y = D_j^{-1}w$$

and

$$c_{i,j+1} = c_{ij} - 2w_i y$$

for all i with $\alpha_{ij} = 0$. Set $\hat{c}_{\nu j} = c_{\nu,j+1}$ and go to Step 2.3.

Step 2.2

Compute

$$\lambda_i = a_k' c_{ij} \quad \text{for all } i \text{ with } \alpha_{ij} > 0 .$$

If $\lambda_i \leq 0$ for all i, stop with the message that problem (2.1) has no feasible solution; otherwise determine the smallest ν such that

$$\frac{u_{\nu j}}{\lambda_\nu} = \min\left\{\frac{u_{ij}}{\lambda_i} \;\middle|\; \text{for all } i \text{ with } \alpha_{ij} > 0 \text{ and } \lambda_i > 0\right\} .$$

Set $\hat{c}_{\nu j} = c_{\nu j}$, replace

$$u_{\nu j} \text{ with } \frac{u_{\nu j}}{\lambda_\nu}$$

and

$$u_{ij} \text{ with } u_{ij} - \frac{u_{\nu j}}{\lambda_\nu}\lambda_i$$

for all $i \neq \nu$ with $\alpha_{ij} > 0$. Go to Step 2.3.

Step 2.3

Compute

$$c_{\nu,j+1} = \frac{\hat{c}_{\nu j}}{a_k' \hat{c}_{\nu j}}$$

and

$$c_{i,j+1} = c_{ij} - (a_k' c_{ij}) c_{\nu,j+1}$$

for all $i \neq \nu$ with $\alpha_{ij} > 0$. Set

$$D_{j+1}^{-1} = (c_{1,j+1},\ldots,c_{n,j+1}), \quad J(x_{j+1}) = \{\alpha_{1,j+1},\ldots,\alpha_{n,j+1}\} ,$$

where $\alpha_{\nu,j+1} = k$ and $\alpha_{i,j+1} = \alpha_{ij}$ for all $i \neq \nu$. Go to Step 3.

Step 3

Compute

$$s_j = (a_k^t x_j - b_k)c_{\nu,j+1}$$

and

$$\delta_{ij} = -c_{i,j+1}^t C s_j \quad \text{for all} \quad i \quad \text{with} \quad \alpha_{ij} > 0 .$$

If $\delta_{ij} \leq 0$ for all i, set $\sigma_j^* = \infty$; otherwise determine the smallest ℓ such that

$$\sigma_j^* = \frac{u_{\ell j}}{\delta_{\ell j}} = \min\left\{\frac{u_{ij}}{\delta_{ij}} \mid \text{for all } i \text{ with } \alpha_{ij} > 0 \text{ and } \delta_{ij} > 0\right\} .$$

Set

$$\sigma_j = \min\{1, \sigma_j^*\}, \quad x_{j+1} = x_j - \sigma_j s_j ,$$

$$u_{i,j+1} = u_{ij} \quad \text{for all} \quad i \quad \text{with} \quad \alpha_{ij} = 0 ,$$

$$u_{i,j+1} = u_{ij} - \sigma_j \delta_{ij} \quad \text{for all} \quad i \quad \text{with} \quad \alpha_{ij} > 0 .$$

Replace j with $j + 1$. If $\sigma_{j-1} = 1$, go to Step 1. If $\sigma_{j-1} < 1$, go to Step 4.

Step 4

Compute

$$c_{\ell,j+1} = \frac{c_{\ell j}}{\sqrt{c_{\ell j}^t C c_{\ell j}}},$$

$$c_{i,j+1} = c_{ij} - (c_{\ell,j+1}^t C c_{ij})c_{\ell,j+1} \quad \text{for all} \quad i \neq \ell \quad \text{with} \quad \alpha_{ij} > 0 ,$$

$$c_{i,j+1} = c_{ij} \quad \text{for all} \quad i \quad \text{with} \quad \alpha_{ij} = 0 .$$

Set

$$D_{j+1}^{-1} = (c_{1,j+1}, \ldots, c_{n,j+1}), \quad J(x_{j+1}) = \{\alpha_{1,j+1}, \ldots, \alpha_{n,j+1}\} ,$$

where $\alpha_{\ell,j+1} = 0$ and $\alpha_{i,j+1} = \alpha_{ij}$ for all $i \neq \ell$. Replace j with $j + 1$ and go to Step 3.

The following lemma establishes the properties of the matrices D_j and D_j^{-1} which guarantee the finite termination of the algorithm.

Lemma

Define the set I such that $j \in I$ if and only if the algorithm uses Step 1 at the j-th iteration. Let

$$D_j^{-1} = (c_{1j},\ldots,c_{nj}),\ J(x_j) = \{\alpha_{1j},\ldots,\alpha_{nj}\},\quad x_j,\ \text{and}\ u_j$$

be as defined by the algorithm and denote the columns of D_j^t by $d_{1j},\ldots,d_{nj}$. Then, for every $j \in I$, we have the following properties

i) $d_{ij} = Cc_{ij}$ for all i with $\alpha_{ij} = 0$,

ii) $d_{ij} = a_{\alpha ij}$ for all i with $\alpha_{ij} > 0$,

iii) $a_{\alpha ij}x_j = b_{\alpha ij}$ for all i with $\alpha_{ij} > 0$,

iv) $-(c + Cx_j) = D_j^t u_j$ with $u_{ij} = 0$ for all i with $\alpha_{ij} = 0$ and $u_{ij} \geq 0$ for all i with $\alpha_{ij} > 0$.

Proof:

For $j = 0$ the statements of the lemma follow from the definition of D_0^{-1} and $J(x_0)$. Let $j \in I$ and assume that i)-iv) are true. First we consider the case that $\alpha_{ij} = 0$ for at least one i and that z_ν, as defined in Step 2.1, is different from zero. Let w be as defined in Step 2.1 of the algorithm and set

$$H = I - 2ww' .$$

Then H is a Householder-matrix with the property (see e.g. [3])

$$Hz = \delta\|z\|e_\nu . \tag{4.1}$$

Furthermore, let

$$\hat{D}_j^{-1} = D_j^{-1}H = (\hat{c}_{1j},\ldots,\hat{c}_{nj}) .$$

Then

$$\hat{D}_j^t = D_j^tH = (\hat{d}_{1j},\ldots,\hat{d}_{nj}) .$$

By definition,

$$\hat{c}_{ij} = c_{ij} - 2w_iD_j^{-1}w = c_{ij} - 2w_iy,\quad i = 1,\ldots,n .$$

Since $w_i = z_i = 0$ for all i with $\alpha_{ij} > 0$, we have

$$\hat{c}_{ij} = c_{ij} \quad \text{for all } i \text{ with } \alpha_{ij} > 0$$

and

$$\hat{c}_{ij} = c_{i,j+1} \quad \text{for all} \quad i \quad \text{with} \quad \alpha_{ij} = 0 \ ,$$

where the vectors $c_{i,j+1}$ are as defined by Step 2.1 of the algorithm. Similarly,

$$\hat{d}_{ij} = d_{ij} = a_{\alpha ij} \quad \text{for all} \quad i \quad \text{with} \quad \alpha_{ij} > 0$$

and

$$\hat{d}_{ij} = d_{ij} - 2w_i D_j w \quad \text{for all} \quad i \quad \text{with} \quad \alpha_{ij} = 0 \ .$$

Since $d_{ij} = Cc_{ij}$ for all i with $\alpha_{ij} = 0$ and $w_i = 0$ for all i with $\alpha_{ij} > 0$, it follows that, for all i with $\alpha_{ij} = 0$,

$$\begin{aligned} \hat{d}_{ij} &= Cc_{ij} - 2w_i C D_j^{-1} w \\ &= Cc_{ij} - 2w_i C y \\ &= C(c_{ij} - 2w_i y) \\ &= C\hat{c}_{ij} \ . \end{aligned}$$

This shows that $\hat{D}_j$, x_j, and $\hat{u}_j = u_j$ have the properties i)-iv). Furthermore let $a_k = \hat{a}_k + \tilde{a}_k$, where $\hat{a}_k \in \text{span}\{d_{ij} \mid \text{all } i \text{ with } \alpha_{ij} > 0\}$ and $\tilde{a}_k \in \text{span}\{d_{ij} \mid \text{all } i \text{ with } \alpha_{ij} = 0\}$. Then

$$z' = \tilde{a}_k' D_j^{-1} \quad \text{and} \quad \hat{a}_k' c_{ij} = 0 \quad \text{for all} \quad i \quad \text{with} \quad \alpha_{ij} = 0 \ .$$

Therefore, it follows from (4.1) that

$$\tilde{a}_k' \hat{D}_j^{-1} = \tilde{a}_k' D_j^{-1} H = z' H = \delta \|z\| e_\nu' \ ,$$

which implies that

$$\hat{a}_k' \hat{c}_{ij} = \begin{cases} 0 & \text{for all} \quad i \neq \nu \quad \text{with} \quad \alpha_{ij} = 0 \\ \delta \|z\| & \text{for} \quad i = \nu \ . \end{cases}$$

If ν is determined by Step 2.2, we set $\hat{D}_j^{-1} = D_j^{-1}$. In either case we have

$$\hat{a}_k' \hat{c}_{ij} = 0 \quad \text{for all} \quad i \neq \nu \quad \text{with} \quad \alpha_{ij} = 0 \ . \tag{4.2}$$

Let D_{j+1}^i be obtained from $\hat{D}_j^i$ by replacing the ν-th column with a_k. Then it follows from (4.2) that the columns $c_{i,j+1}$ of D_{j+1}^{-1} are as defined in Step 2.1 and 2.2 of the algorithm. If we set $\hat{u}_j = u_j$ when ν is determined by Step 2.1 and equal to the vector defined in Step 2.2 otherwise, it is easy to verify that D_{j+1}^i, D_{j+1}^{-1}, x_j, $\hat{u}_j$, and $J(x_{j+1})$ have properties i)-iv) with the exception that

$$a_{\alpha\nu,j+1}' x_j > b_{\alpha\nu,j+1} \ .$$

Now let σ_j, x_{j+1}, and u_{j+1} be as determined by Step 3. Then $j+1 \in I$ if and only if $\sigma_j = 1$ in which case $a'_{\alpha\nu,j+1}x_{j+1} = b_{\alpha\nu,j+1}$, i.e. properties i)-iv) hold. If $\sigma_j < 1$, let D_{j+2} be the matrix obtained from D_{j+1} by replacing the ℓ-th column with $c_{\ell,j+1}/\sqrt{c'_{\ell,j+1}Cc_{\ell,j+1}}$. Since $c'_{\ell,j+1}Cc_{i,j+1}$ for all i with $\alpha_{i,j+1} = 0$, the columns of D_{j+2}^{-1} are as determined by Step 4. Furthermore, the ℓ-th component of u_{j+1} is zero. Thus it follows again that D_{j+2}, D_{j+2}^{-1}, $J(x_{j+2})$, x_{j+1}, and u_{j+1} have properties i)-iv) with the exception that

$$a'_{\alpha\nu,j+2}x_{j+1} > b_{\alpha\nu,j+2} .$$

This completes the proof of the lemma.

The main properties of the algorithm are summarized in the following theorem.

Theorem

Let $x_0, x_1, \ldots$ be the vectors determined by the algorithm.

i) For every j the set

$$\{x_j, x_{j+1}, \ldots, x_{j+n}\}$$

contains at least one pseudo-stationary point.

ii) If x_{j_1} and x_{j_2} are two pseudo-stationary points with $j_1 < j_2$, then

$$Q(x_{j_2}) > Q(x_{j_1}) .$$

iii) The algorithm terminates after a finite number of iterations with either an optimal solution or the information that the given problem has no feasible solutions.

Proof:

i) Let the set I be as defined in the lemma. Then it follows that, for every $j \in I$, x_j is a pseudo-stationary point. At every iteration j with $j \notin I$, the number of positive elements in the set $J(x_j)$ decreases by 1. Hence, there are at most n consecutive iterations such that $j \notin I$.

ii) With $s_j = (a'_k x_j - b_k)c_{\nu,j+1}$ and $\delta_{ij} = -c'_{i,j+1}Cs_j$ we have

$$Cs_j = -\sum \delta_{ij} a_{\alpha i,j+1} , \tag{4.3}$$

where the summation is over all i such that $\alpha_{i,j+1} > 0$. Multiplying (4.3) by s_j we obtain

$$s_j' C s_j = -\delta_{\nu j} a_k' s_j = -\delta_{\nu j}(a_k' x_j - b_k) .$$

Thus $\delta_{\nu j} \leq 0$ and $u_{\nu,j+1} = u_{\nu j} - \delta_{\nu j} \geq 0$. Furthermore,

$$Q(x - \sigma s_j) - Q(x_j) = -\sigma(c + Cx_j)s_j + \frac{\sigma^2}{2} s_j' C s_j$$

$$= \sigma \sum_{\alpha i, j+1} u_{ij} a_{\alpha i, j+1}' s_j + \frac{\sigma^2}{2} s_j' C s_j$$

$$= \sigma u_{\nu j} a_k' s_j + \frac{\sigma^2}{2} s_j' C s_j$$

$$> 0 \quad \text{for} \quad \sigma > 0 .$$

iii) If the algorithm terminates with x_j in Step 1, then $x_j \in R$ and $j \in I$. Thus it follows from the lemma that x_j is a pseudo-stationary point. Because $x_j \in R$ the optimality conditions are satisfied and x_j is indeed an optimal solution. Suppose the algorithm terminates with Step 2.2. Then

$$a_k = \sum_{\alpha ij > 0} \lambda_i a_{\alpha ij} \quad \text{with} \quad \lambda_i \leq 0 .$$

Let x be such that $a_{\alpha ij}' x \leq b_{\alpha ij}$ for all i with $\alpha_{ij} > 0$. Set $s = x_j - x$. Then $a_{\alpha ij}' s \geq 0$ for all i with $\alpha_{ij} > 0$. Thus

$$a_k' x = a_k'(x_j - s) = a_k' x_j - \sum_{\alpha ij > 0} \lambda_i a_{\alpha ij}' s \geq a_k' x_j > b .$$

This shows that $R = \emptyset$. Since there are only finitely many pseudo-stationary points, it follows from part ii) of the theorem that the algorithm terminates after a finite number of iterations.

REFERENCES

[1] Best, M. J. and K. Ritter, "A quadratic programming algorithm", Technical Summary Report, University of Wisconsin-Madison, to appear.

[2] Mangasarian, O. L., "Nonlinear Programming", McGraw-Hill, 1969.

[3] Noble, B., "Applied Linear Algebra", Prentice Hall, 1969.

KR:scr

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 2733	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A DUAL QUADRATIC PROGRAMMING ALGORITHM		5. TYPE OF REPORT & PERIOD COVERED Summary Report - no specific reporting period 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Klaus Ritter		8. CONTRACT OR GRANT NUMBER(s) DAAG29-80-C-0041
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematics Research Center, University of Wisconsin 610 Walnut Street Madison, Wisconsin 53706		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit Number 5 - Optimization and Large Scale Systems
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office P.O. Box 12211 Research Triangle Park, North Carolina 27709		12. REPORT DATE August 1984 13. NUMBER OF PAGES 19
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Quadratic programming
Duality
Optimization
Conjugate directions

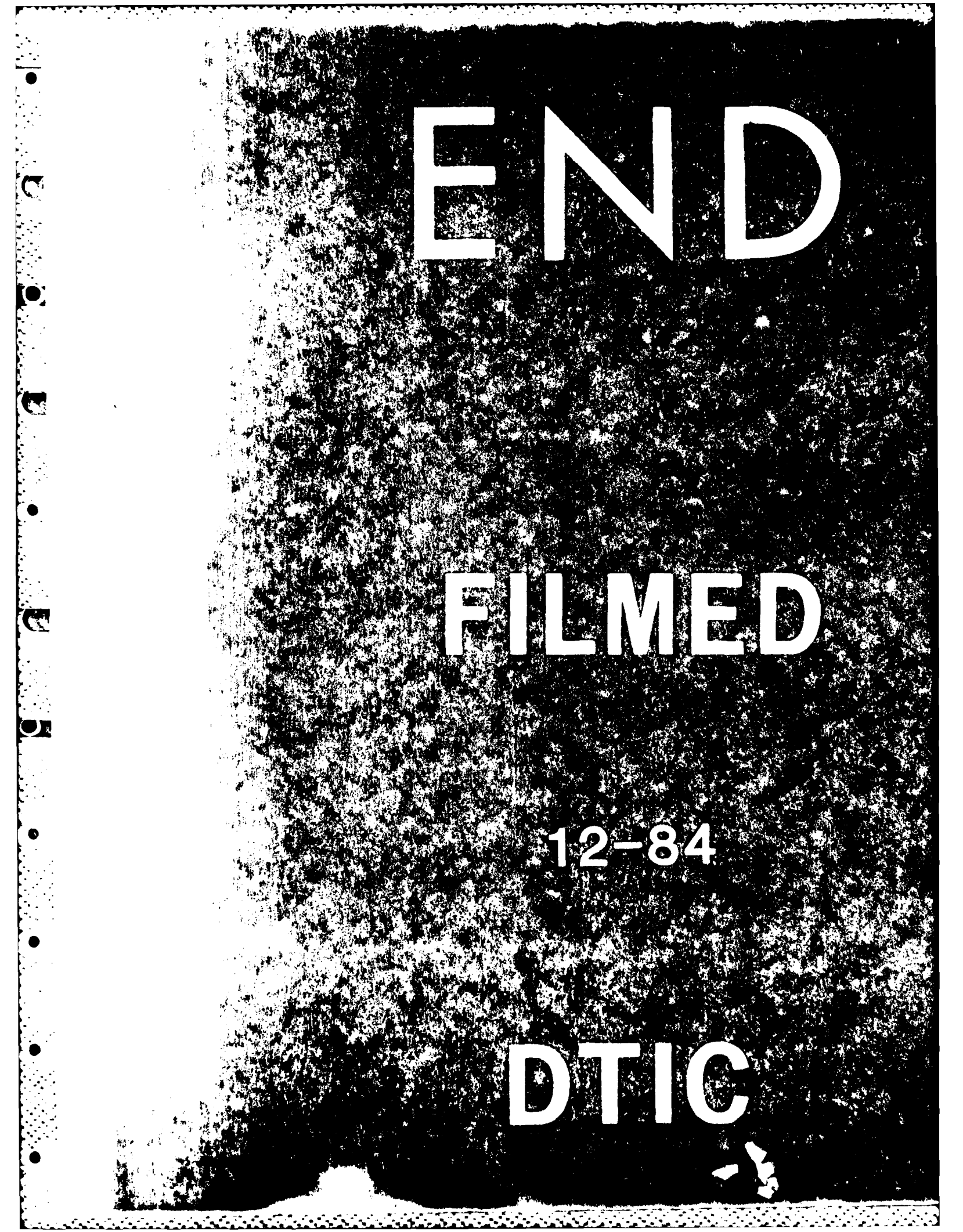
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

By using conjugate directions a method for solving convex quadratic programming problems is developed. The algorithm generates a sequence of dual feasible solutions and terminates after a finite number of steps.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)


END
FILMED
12-84
DTIC